

1 **Appendix B**

2

3 **Concepts used in an Exemplary Object Model or Object Class Definitions**

4

5 For the purpose of describing different concepts, the examples of (Java) class definitions
6 given in the Figure 19 use Java. These classes have been defined using the sample
7 database 'pubs' which comes with Microsoft SQL Server relational database
8 management system.

9

10 **Primitive and complex attributes**

11 A primitive attribute of a class is an instance variable of simple (primitive) type. For
12 example, attributes of type integer, String, boolean are primitive attributes. In the
13 Figure 19, title_id and ytd_sales of class Title are primitive attributes.

14 A complex attribute is a reference to an object (or a collection of objects) of another
15 class. For example pubInfo and titles of class Publisher are complex attributes.

16 In general, all instances of a class are stored in the same relational table. Primitive
17 attributes are stored in the columns of that class. The appropriate SQL data type of the
18 column may be different However, a complex attribute is not (cannot be) stored in the
19 same table. Instead, the object(s) referenced by a complex attribute is stored in the
20 table of the corresponding class. To subsequently retrieve those referenced objects as
21 part of the containing object, a set of attributes of the containing object is used (as
22 foreign key) to locate them. If a complex attribute is referencing a collection of objects,
23 their ordering may be specified.

1

2 **Containment by value**

3 All primitive attributes are contained by value in an object. That is, they cannot exist
4 independent of the object. Objects referenced by a complex attribute that cannot
5 logically exist without the existence of the referencing (containing) object are also said
6 to be contained by value. If the containing object is deleted, the contained by value
7 objects should also be deleted. pubInfo and titles are examples of attributes contained
8 by value in the class Publisher.

9

10 **Containment by reference**

11 A complex attribute which points to an object which may exist independent of the
12 existence of the referencing (containing) object is an attribute contained by reference.
13 If the referencing (containing) object is deleted, the referenced object need not be
14 deleted. The publisher attribute of Emp class is contained by reference.

15

16 **Collection class**

17 A collection class is a pseudo-class describing a certain type (array or vector) of
18 collection of objects of a given class. A collection class has an optional notion of
19 ordering of elements of the collection. Collection class specification is needed to
20 describe those complex attributes which reference a collection (array or vector) of
21 objects. This is needed only for object relational mapping specification; no explicit
22 class is defined at the language level.

1

2 **Shallow and Deep queries**

3 When an object is normally retrieved from the database, it is retrieved with all the
4 complex attributes instantiated with referenced objects (recursively). This is called
5 deep query. If none of the referenced objects are retrieved (i. e. only the primitive
6 attributes are retrieved) , it is called a shallow query.

7

8 **Directed operation options**

9 Sometimes, just deep or shallow options may not be the most optimal way of doing an
10 operation (query, insert, update, delete). For example, if an object has 5 complex
11 (reference) attributes and we are interested in getting just 2 of them, how do we specify
12 that? The answer is directed options. Directed options further qualify the depth of the
13 operation in one of the following ways: 1) if the overall operation is deep, one may
14 specify a list of class and attribute name pairs such that all except those complex
15 attributes should be followed and 2) if the overall operation is shallow, one may specify
16 a list of class and attribute name pairs such that only those complex attributes should be
17 followed. Depending upon the application needs at certain stage of processing, these
18 options may be employed to achieve efficiency or some semantic requirements.

19

20 **Database URL**

21 A database URL contains the URL (Universal Resource Locator) understood by the
22 appropriate JDBC driver. Additionally, it has user name and password components.

- 1 Optionally, it has information about Object-Relational Mapping Id (ORMId) and/or
- 2 Object-Relational Mapping File (ORMFile). They have been explained further under
- 3 the grammar rules of <DATABASE-URL> in a subsequent section.
- 4